VTT Technical Research Centre of Finland

The Design Challenges of Drone Swarm Control

Saffre, Fabrice; Hildmann, Hanno; Karvonen, Hannu

Published in:
Engineering Psychology and Cognitive Ergonomics, HCII 2021

DOI:
10.1007/978-3-030-77932-0_32

Published: 01/01/2021

Document Version
Early version, also known as pre-print

The Design Challenges of Drone Swarm Control*

Fabrice Saffre[1][0000-0003-1194-5408], Hanno Hildmann[2][0000-0002-1471-5774], and Hannu Karvonen[3][0000-0002-8838-7101]

[1] VTT – Technical Research Centre of Finland Ltd, MIKES bldg, Tekniikantie 1, 02150 Espoo, Finland `fabrice.saffre@vtt.fi`

[2] TNO – Netherlands Organisation for Applied Scientific Research, Oude Waalsdorperweg 63, 2597 AK, Den Haag `hanno.hildmann@tno.nl`

[3] VTT – Technical Research Centre of Finland Ltd, FutureHub bldg, Tekniikantie 21, 02150 Espoo, Finland `hannu.karvonen@vtt.fi`

Abstract. It is widely recognised that swarms are the likely next step for Unmanned Aerial Vehicle (UAV) or drone technology. Although substantially increased autonomy for navigation, data collection and decision-making is very much part of the *"collective artificial intelligence"* vision, this expected development raises questions about the most productive form of interaction between the swarm and its human operator(s). On the one hand, low-level *"micro-management"* of every unit clearly nullifies many of the advantages of using swarms. On the other, retaining an ability to exercise some control over the swarm's objectives and real-time behaviour is obviously paramount. We present two families of control methods, direct and indirect, that we believe could be used to design suitable, i.e. simultaneously intuitive, easy to use, powerful and flexible, Graphical User Interfaces (GUI) that would allow a single operator to choreograph a swarm's actions. Simulation results are used to illustrate the concept and perform a quantitative performance analysis of both control methods in different scenarios. Human factors aspects related to drone swarm control are identified and both control methods are discussed from the human operator's usage point of view. We conclude that the direct approach is more suitable over short time-scales (*"tactical"* level), whilst indirect methods allow to specify more abstract long-term objectives (*"operational"* level), making them naturally complementary.

Keywords: Drone Swarms · Direct Control · Indirect Control · Human Factors · Autonomous Systems · Self-Organisation.

1 Introduction

Drones (or unmanned aerial vehicles, UAVs) are increasingly becoming an everyday tool [9] in a growing number of application areas [17], as a convenient and

* The Academy of Finland is acknowledged for financial support of project '`Finnish UAV Ecosystem`' 337878 (FUAVE, project grant number 337878). This research was also partly funded by the Scientific Advisory Board for Defence (MATINE) in the `CounterSwarm` project.

cost-efficient way of remote sensing [4] or to gather information best acquired from a vantage point not easily accessible by other means (e.g., [18, 2]).

In the vast majority of cases, these benefits are achieved by combining remote control by a human operator with relatively simple autonomous features, such as attitude control [22, 24], obstacle avoidance, and return-to-base functions. In the case of beyond visual line of sight (BVLOS) operations, the assumption is often that the human in the loop uses a real-time video feed to pilot the drone as if he/she were on board (see Figure 1).

Fig. 1. A TNO reconnaissance drone departing for BVLOS flight during the (now concluded) EU funded ALFA (Advanced Low Flying Aircrafts Detection and Tracking) project [5] where it facilitates automatic threat evaluation for border security and surveillance. The pilot uses the control interface to *see* what the drone can see.

However, another paradigm is gaining momentum that is poised to challenge this mode of operation: fleets or swarms of drones. Simple common sense suggests that it is possible to achieve more with several units working as a team than with a single UAV and, accordingly, that an ability to leverage drone swarms would have a multiplicative effect on the usefulness of the technology. However, this trivial statement hides the considerable underlying complexity of orchestrating or choreographing the joint operation of a collective dozens or hundreds strong.

In the case of exploration or surveillance, for instance, it is obvious that n identical drones have the potential to cover an area roughly n times larger in the same amount of time as a single unit (notwithstanding restrictions imposed by battery life or shared point of origin). However, this implicitly assumes division of labour, i.e., no or limited overlap between flight paths, to avoid the duplication of effort. Even in this simplest of cases and without any real-time change of objectives, this would require assigning each drone to a particular zone in the area of interest. Furthermore, short of having one human operator per drone, each one would have to fly its respective patrol route autonomously, without real-time supervision. This in itself poses various safety concerns [20].

In the area of drone swarms, there are a number of concrete research questions [9]. For example, what interface design would best allow the user to monitor and operate a drone swarm? Furthermore, is there a suitable trade-off between micro-management (i.e., directly piloting every unit in real-time) and assigning global objectives to the entire fleet? And if the answer to the previous question is 'yes', what autonomous features are required and what kind of useful collective behaviour is possible? Finally, what control functions would allow to the operator of a drone swarm to achieve maximum utility for minimum complexity?

As a starting point, it does seem useful and necessary to create a *"taxonomy"* of control functions and associated distributed algorithms for drone swarms. A first distinction would be between those that involve directly piloting a subset of units (possibly a single one) and those that instead specify abstract collective goals. It is worth noting that these two approaches, although functionally different, are not mutually exclusive and could govern different aspects of fleet operations in parallel / at the same time.

The former (direct control) could be used on a short time-scale to coordinate the movement of the swarm into a chosen direction through formation flying. In this scenario, the human operator pilots the remote-controlled *"leader"* unit and the other members of the swarm use simple autonomous features (relative positioning) to spread out around it, forming a pattern. This would result in the type of collective behaviour most often cited as an example of swarming, i.e., flocking. More advanced functions could be introduced in the form of a simple ability to update key parameter values in real-time. For instance, how tight or loose the formation is could easily be controlled by fine-tuning the separation distance. Other basic commands could involve, e.g., instructions to follow, spread around or align with the leader perpendicularly to the direction of movement. Similar commands could be used to control altitude, determining, for instance, whether *"subordinate"* units should distribute themselves in the same horizontal plane as the leader or form a 3D lattice (with upper and lower bounds).

The latter (indirect control through abstract collective goals) is likely to be more useful on a longer time-scale and/or when the swarm is expected to fulfil its mission without direct human control or supervision. For instance, the drone *"colony"* could be tasked with patrolling a region of interest over an extended period (hours, days, or more). This would require much more complex autonomous features in the form of decentralised resource-management and collective decision-making to ensure that the airborne contingent balances the need to recharge with that to visit every part of the target area regularly. In this scenario, the challenge is to create an interface that allows the user to specify and subsequently update such abstract goals intuitively. For example, the ability to *"paint"* a region of arbitrary size and shape on a digital map to designate it as being of interest and communicate this information to the swarm would be paramount. In both direct and indirect control, human factors aspects need to be considered in the design of the system and its related user interfaces.

2 Human Factors Aspects in Drone Swarm Control

As has occurred in many other highly automated domains (e.g., industrial process control, ship navigation, and traditional aviation), conducting drone operations will also eventually evolve to become more of a supervisory task than an active manual control. Consequently, similar human factors problems that have been identified in these other domains will then become prevalent. These issues include, but are not limited to, operator trust in automation [23], excessive mental workload in exception situations [25], situation awareness issues [12, 13], operator boredom [11], work vigilance [6] and the integrity of the artificial intelligence [26]. Additionally, with higher automation levels supported with AI, the supervisory control of multiple UAVs will become possible. To approach the human factors problems of controlling a drone swarm, various approaches have been discussed in the literature, only a fraction of which is discussed here.

To analyse, optimise and divide the tasks to be conducted by humans and by the AI/automated system in drone swarm control, task/work analyses are a key approach. They are especially suitable to consider the human limitations and support the definition of human operator's meaningful tasks and decision-making in the design phase of the system [1]. As a one guideline for design when considering human-automation task allocation in drone swarm control, it has been found out that instead of *management by consent* (automation as an assistant to the operator), *management by exception* improves the operator's performance [10]. According to [16], management by exception means that the automation decides to take an action based on some set of predetermined criteria and gives operators only a chance to veto the automation's decision.

In systems engineering, defining a Concept of Operations (ConOps) for the drone swarm control in the start of the system design is essential [14]. From the human factors point of view, this ConOps typically includes a clear description of, for example, the division of tasks between human and the automation, operator tools, roles/responsibilities, and procedures [3]. This ConOps should work as a boundary object (see, e.g., [29]) and allow the relevant stakeholders (e.g., engineers, users, and designers) to discuss about the system under development and the related aims in a manner understandable for all involved parties.

On a methodological level, for example, ecological interface design (EID) has been successfully applied to the control of UAV swarms by [15]. The results of this study showed that EID-inspired interface design enabled operators to control a drone swarm and successfully resolve failures during mission execution [15]. Particularly, the ecological interface designs promoted creative problem-solving activities to scenarios that could not have been solved by following a fixed procedure (see details in [15]).

Regarding detailed control systems, for example, [19] have provided a review of human-system interface (HSI) solutions for the management of swarms of drones. Their main conclusion from this review was that allowing user and mission-specific customization to user interfaces and raising the swarm's level of autonomy to reduce the cognitive workload shouldered by the operator are beneficial and improve operators' situation awareness [19].

There have also been some studies about different interaction modalities for drone swarm control, such as gesture and touch [21]. However, the benefits of user interfaces with novel input paradigms remains unclear compared to traditional point-and-click interfaces. In general, in drone swarm control HSI design it is important to ensure that the operator has adequate means to first observe and then direct the automation's functioning in order to be responsive to potential situation specific changes [8]. Therefore, both the hardware and software solutions developed to monitor and control the swarm need to be suitable for the specific situation and task at hand without cumbersome interaction solutions that may hinder safe and efficient operations.

3 Direct Control Methods

The most straightforward method for controlling a fleet of semi-autonomous UAVs, at least over short time-scales (of the order of a drone's battery life), is to pilot one or a few participating units directly and use parameterised formation flying to orchestrate the collective behaviour of the rest of the swarm. This "leader" may very well differ from the other members of the *"flock"*. For instance, it could be a special drone with enhanced capabilities (e.g., for reliable long-range communication and telemetry), or a manned aircraft that the swarm is meant to escort or *"extend"* (acting as a network of distributed sensors).

Notwithstanding such specifics, the difficulty lies in identifying the right balance between precise control and ease of use. The human operating the swarm may have limited time or cognitive resources to allocate to this task, as other activities may require his or her urgent attention. Therefore, to minimise mental workload, orchestrating the UAV collective should be as easy and seamless as possible instead of a fastidious exercise with real time fine-tuning. To use a common metaphor: the swarm should *"feel"* like a mere extension of the user's own sensing and actuation capabilities. This aim means that identifying the right parameters, a range of suitable values for them, and a suite of intuitive, user-friendly tools to pick or change one or more of these parameter values is paramount. Clear and illustrative visualisations of the swarm's behaviour as well as *'what if'* scenarios in the user interfaces are also essential for fluent operations.

There is no proven or *"one size fits all"* approach to solving this conundrum. The framework we present here is meant as an illustration of a possible *"swarm interface"* design, not as a final product. Different applications will undoubtedly require specific additional functions that we do not discuss here, as they would require restricting our findings to a particular mission-specific domain.

UAVs, as physical devices, are defined by hardware characteristics that act as constraints over what can and cannot be achieved. Some of these traits, such as, for example, battery life, are of critical importance when considering longer time-scales (days or weeks) but are not particularly relevant for short periods (which we previously argued is the context in which direct control methods are likely to be most useful).

Other limiting factors also play an essential role in formation flying, such as:

- Positioning accuracy
- Maximum speed
- Maximum acceleration
- Sensor/communication range

Here, it is good to note that fixed-wing aircraft represent a special case in that they rely on lift to stay airborne (and so have a minimum speed threshold too) and cannot accelerate in any direction, but rely on course correction. What follows assumes that the swarm is comprised of rotary-wing drones (e.g., quadcopters or (remote controlled or autonomous) helicopters [7]).

Positioning accuracy is a tricky parameter to take into account, but it is mostly relevant for close formations such as the ones used for aerobatics display. In most other applications, the target separation distance between units will be considerably higher (sometimes by over one order of magnitude) than the average positioning error. For instance, GPS is typically accurate down to a few meters, so if drones are attempting to maintain a separation of 50+ meters even based solely on broadcasted GPS coordinates, the error will already not have much impact. Furthermore, other methods such as dead reckoning, radio signal triangulation and attenuation analysis, or even real-time computer vision can be used to improve accuracy of (relative) positioning. In the remainder of this section, the coordinates are assumed to be the ones reported by the UAVs, which means that the actual location might slightly differ, but not to the extent that it will significantly affect the configuration of the swarm.

Speed and acceleration are another matter. Not only do their maximum possible values, imposed by aerodynamic characteristics, represent an upper bound to which instructions given to the swarm must conform, they also less directly but significantly affect flying patterns.

Our simulation results suggest that active/deliberate modulation of the target speed (V) and acceleration (*delta-*V) by the swarm operator can be one of the simplest and most efficient way to control collective behaviour through a single pair of parameters. It is important to understand here that, whereas *maximum* values are hardware constraints, *target* acceleration and speed are not. For instance, a quadcopter may not be capable of accelerating at a faster rate than $10\frac{m}{s^2}$ but there is nothing preventing on-board software from targeting a *lower* value if so instructed. Since lower acceleration means greater inertia, under the influence of the same *"forces"* (e.g., a tropism), different target *delta-*Vs will result in different movement patterns.

If we consider the case of a single rotary wing UAV, acceleration is the result of its tilting into the direction of intended movement, converting some lift into thrust. The angle of the tilt, combined with a possible change in the speed of the rotors/blades, is what determines the value of the *delta-*V. If we hypothesise that the rotors automatically adjust their speed to keep lift (and so altitude) constant by default, then acceleration can be controlled via tilt angle only.

For instance, a drone might have a maximum *"safe"* tilting angle (determined by physical constraints such as the top speed at which rotors can spin) allowing for a maximum horizontal acceleration of $10\frac{m}{s^2}$. At this angle, it will take such a device $1s$ to go from hovering (stationary) to traveling at a target velocity of $10\frac{m}{s}$, during which time it will have travelled $5m$ into the intended direction. It will take another $1s$ (and another $5m$) for it to slow down and return to hovering mode (N.B. these figures assume a negligible drag coefficient).

Let us consider the situation in which a drone traveling East at a cruising speed of $10\frac{m}{s}$ is instructed to change direction and go North, using the maximum allowable acceleration. Such a device will first need to stop its Eastward movement ($1s$, $5m$), then accelerate North (again $1s$, $5m$). The change of course takes $2s$ and the drone will reach its new heading and cruising speed at a point $5m$ East (slowing down) and $5m$ North (reaccelerating) of the position where the instruction was received. Had the tilting angle been such that the acceleration rate was only $5\frac{m}{s^2}$ instead of $10\frac{m}{s^2}$, the same manoeuvre would have taken twice the time ($4s$) and the drift east during deceleration would have been $10m$.

3.1 Two drones interaction

These relatively trivial calculations become more subtle when the intended direction of travel depends on the relative location of multiple UAVs attempting to coordinate their movements to achieve a certain objective (e.g., spatial distribution pattern). Figures 2 and 3 visualize these interaction dynamics.

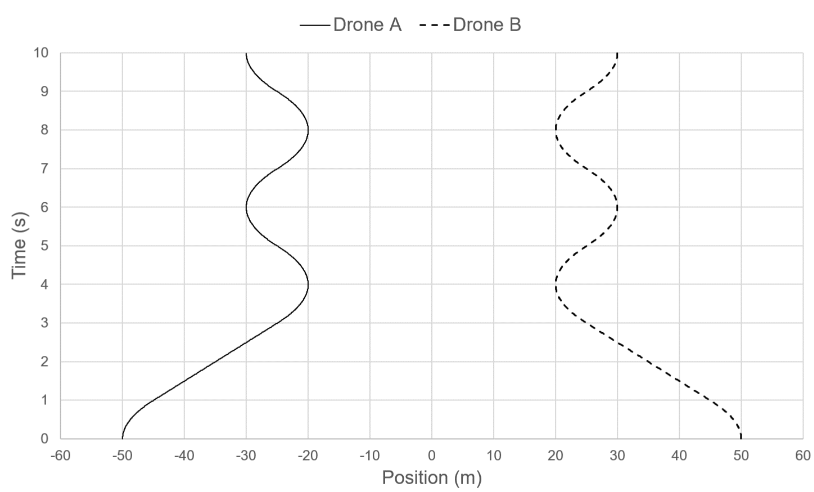

Fig. 2. Visualisation of the two drones interaction dynamics: the evolution of the position over time on the axis the origin of which is the mid-point between the two units (cf. Figure 3 for speed vs. position). The oscillatory regime after $2.5s$ is clearly visible.

For example, if two drones start hovering $100m$ apart and the target separation distance is $50m$, they will accelerate toward each other, say at the maximum allowable *delta-*V $(10\frac{m}{s^2})$. They will reach their cruising speed after $1s$, having travelled $5m$ each, in opposite directions, i.e., $90m$ apart. Another two seconds later, they will have travelled another $20m$ each and be $50m$ apart as instructed. However, at this point, they are flying toward each other at $10\frac{m}{s}$. If they start decelerating immediately, they will be only $40m$ apart by the time they are stationary (overshooting the target separation). Intuitively, this means that they should now accelerate in the opposite direction to open the gap. If they do so, they will be $50m$ from each other again one second later, but traveling at $10\frac{m}{s}$. This will initiate an oscillation around the target separation distance, with an amplitude of $20m$. This is illustrated in Figures 2 and 3.

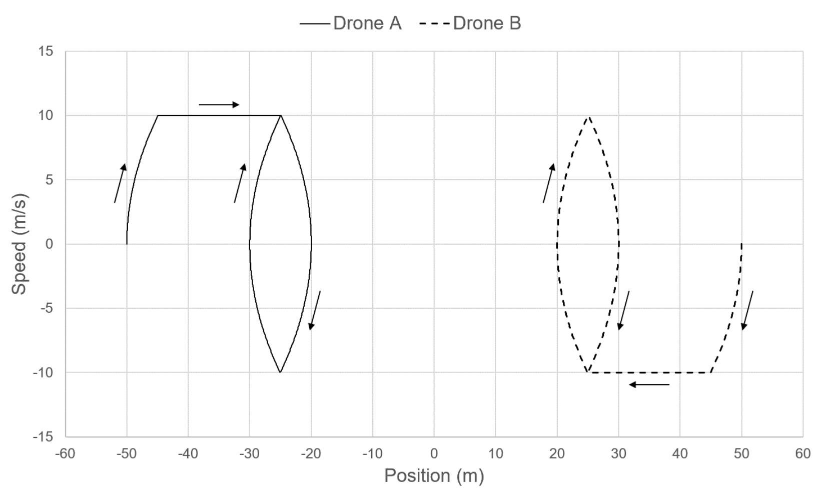

Fig. 3. Visualisation of the two drones interaction dynamics: the speed vs. position (for comparison, cf. Figure 2 for the evolution of the position over time). The *"spindle-shaped"* part corresponds to the oscillation, when the two drones travel back and forth around the equilibrium point. Arrows indicate the passage of time (the closed and indefinitely repeating loop corresponding to the oscillatory regime).

Had they been limited to a tilt angle and acceleration of $5\frac{m}{s^2}$, the scenario would be qualitatively identical but quantitatively different. Indeed, the two UAVs would have reached their cruising speed $(10\frac{m}{s})$ not one but two seconds later and $80m$ apart instead of 90. Having reached the target separation distance $1.5s$ later instead of 2, they would require another two seconds to come to a halt, closing the distance between them to $30m$ (instead of 40). Accelerating away from each other, still at the same reduced rate of $5\frac{m}{s^2}$, then slowing down after reaching the $50m$ separation target, they will be $70m$ apart before they have returned to hovering mode. The oscillation is now double the amplitude, the distance between the two drones varying between $30m$ and $70m$. The perhaps somewhat counter-intuitive conclusion is that a higher acceleration rate results in a reduced deviation from the target separation at steady state.

These effects could of course be counteracted by simply anticipating the overshoot and decelerating pre-emptively, but what would be an easy calculation in the above example (with just two drones traveling along a single dimension) becomes almost impossible in a swarm dozens of units strong, *"pushing and pulling"* each other in 2D or 3D space.

3.2 Multiple drones interaction

We used simulation extensively to investigate the collective dynamics of a swarm controlled by combining direct piloting of a single unit (*"leader"*) with real-time modification of the aforementioned parameter values (target speed, acceleration and separation). The results of these numerical experiments are summarised in this section. We present the user interface (Figure 4) as well as screenshots of the swarm in various configurations together with the corresponding parameter settings (in Figures 6 and 7), to emphasise the link between them.

In order to ensure reproducibility of our findings, we must first disclose the *"hidden"* rules of interaction, i.e., those that are, so to speak, *"hard-coded"* into the rules governing drone behaviour and not tied to any modifiable parameter. It is important to understand that these rules are ad-hoc in nature and that we make no claim of having investigated them in any meaningful way: they were simply found to be suitable to illustrate the type of collective activity pattern that the end user can elicit in the swarm by altering the value of the controllable parameters. The tunable parameters used are shown in Figure 4.

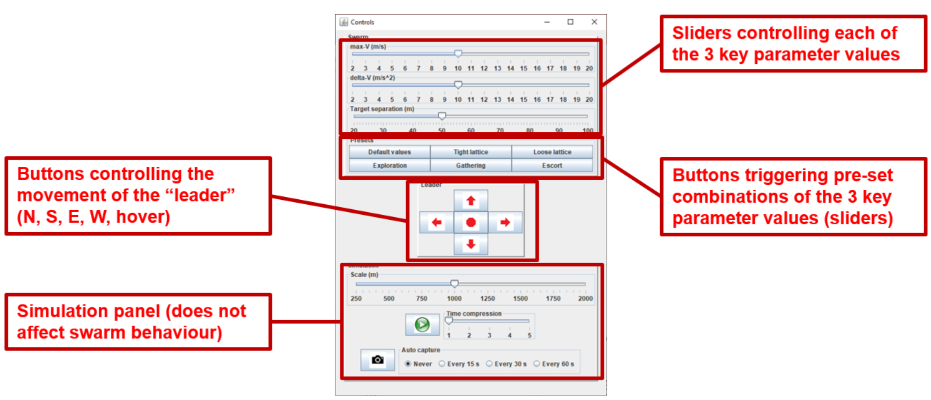

Fig. 4. The user interface with the tunable parameters. Figures 6 and 7 show the sliders to indicate the varied parameters and their respective settings. This could be regarded as an early prototype for a real-world user interface, with the exception of the *"Leader"* sub-panel, which is a very crude approximation for an actual remote-control station.

We simulated a group of 37 drones, one of which is assumed to be remote-controlled and the other 36 semi-autonomous (in the sense that they plan and execute their own movement based on interaction rules and parameter values). The remote-controlled unit (swarm *"leader"*) depends entirely on the human pilot's instructions to perform any action other than hovering in position.

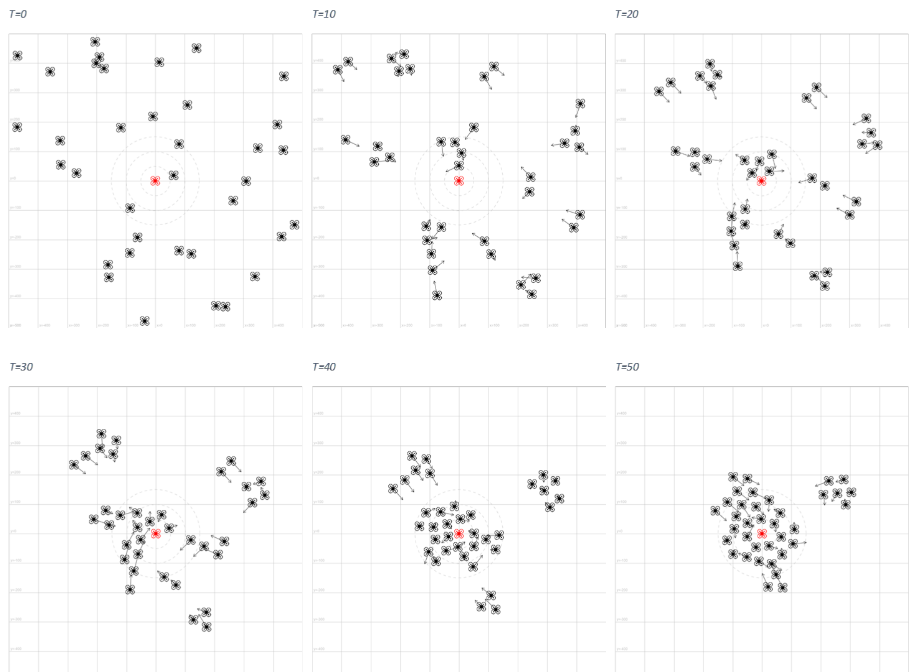

Fig. 5. Clustering process for the default parameter values ($maxV = 10\frac{m}{s}$, $delta\text{-}V = 10\frac{m}{s^2}$, separation $= 50m$) from random initial locations (top-left corner). Screenshots are 10 seconds apart. Arrows are the speed vectors; the screen / display is always centered on the swarm leader, indicated in red.

We hypothesise that all drones are constantly broadcasting their location and that they can all reliably communicate with each other (so every member of the swarm can potentially use the location of any other as input for planning its own movement). In the chosen rule-set, every drone is only using the location of the leader and of its two nearest neighbours (the identity of the two nearest neighbours may of course change over time as the swarm reconfigures itself).

The location of these three other units influences path planning as follows:

- Each one exerts a force inversely proportional to the distance (i.e., $\frac{1}{r}$).
- For distances < target separation this is repulsive, otherwise attractive.
- If the distance falls below half the target separation, this force is multiplied by 10 and a flag is raised (similar to a collision avoidance mechanism).
- The three attraction/repulsion vectors are then added up and the resulting vector is normalised (indicating the direction of acceleration, not intensity).
- This vector is then multiplied by the chosen acceleration rate (*delta-V*) unless the *"collision avoidance"* flag is raised, in which case the maximum value is always used ($20\frac{m}{s^2}$ in our experiment). NB: because of the $\frac{1}{r}$ rule and of the multiplicative factor applied to the repulsion force exerted by a neighbour less than half the target separation away, when in danger of collision, this vector tends to point directly away from the nearest neighbour.

The resulting acceleration vector is then used to update the drone's airspeed, until the target velocity is reached. This is done synchronously (i.e., all speeds and positions are updated simultaneously), with an integration step of $0.01s$.

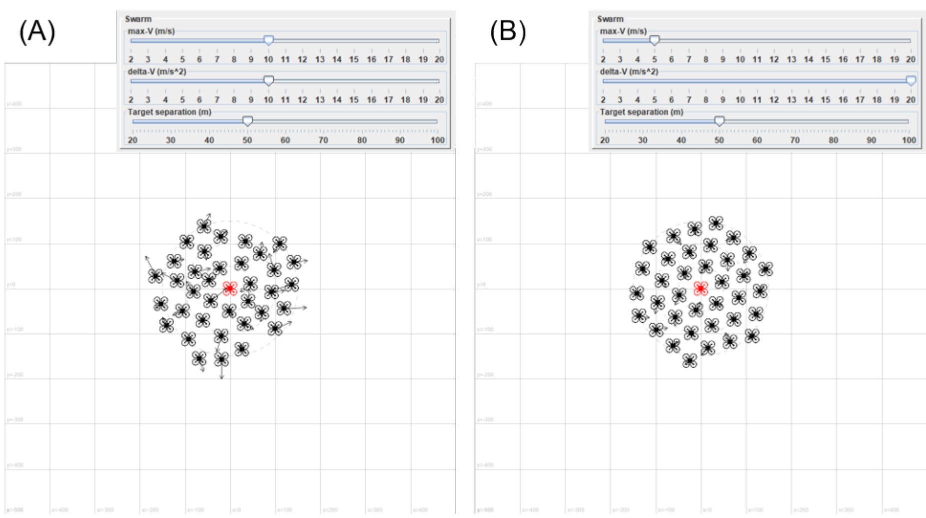

Fig. 6. Controlling the swarm through global parameters only. Each screenshot is a typical illustration of the type of distribution pattern observed at steady state for different combinations of values. Shown are **(A):** *"Flocking"*, **(B):** tight regular lattice.

Unless specified otherwise, units start at a random location within one square kilometre centred on the leader (coordinates origin $x = 0$, $y = 0$), with the only constraint that no two drones can be closer to each other than half the default target separation ($25m$). The default values for acceleration and maximum (or target) speed are $10\frac{m}{s^2}$ and $10\frac{m}{s}$. The sequence shown in Figure 5 illustrates the typical clustering process taking place if no actions are taken by the operator (i.e., all parameters are at their default value and the leader remains stationary).

As illustrated in Figures 6 and 7 it is possible to control the behaviour of the swarm by changing the values of the three previously identified key parameters for relative movement and positioning (*maxV*, *delta-V* and separation).

Some changes will result in a *"quantitative"* difference (e.g., tighter or looser lattice, cf. 6(B), 7(A)) others may bring about a *"qualitative"* change (e.g., from holding position, 7(A), to adopting a random search pattern, 7(B)).

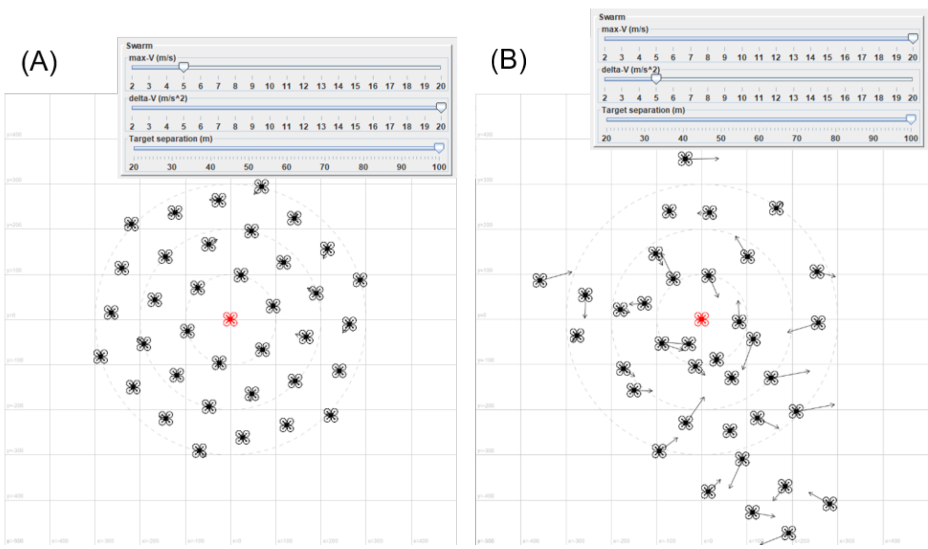

Fig. 7. Controlling the swarm through global parameters only (continuation of Figure 6). Each screenshot is a typical illustration of the type of distribution pattern observed at steady state for different combinations of values. Shown are **(A):** loose regular lattice, **(B):** *"Exploration"*. The swarm will continously self-organize and transition, seemingly spontaneously, between these states as the human operator updates parameter value(s).

Figures 6 and 7 show typical examples. The precise influence of each parameter can be analysed using statistical methods, which should inform the design of the graphical user interface (e.g. by identifying suitable maximum and minimum parameter values). Such principled investigation is also critical to discover possibly *"unsafe"* combinations (e.g. if the ratio between *maxV* and *delta-V* is such that it increases the risk of collision beyond an acceptable level).

To illustrate this approach we present an analysis of a numerical experiment designed to study how changing the acceleration rate (*delta-V*) affects the swarm's ability to form and maintain a regular lattice. Figure 8, shows the resulting frequency distribution.

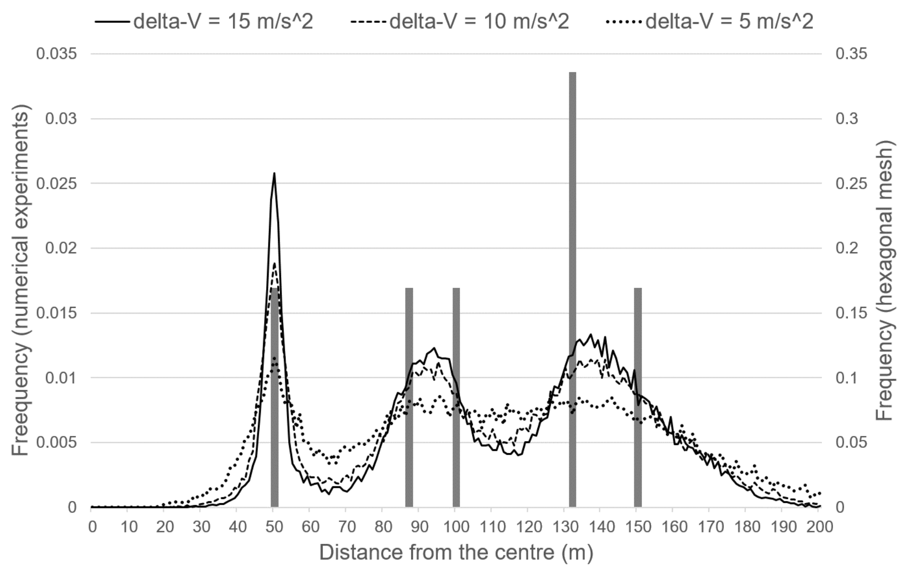

Fig. 8. Frequency distribution of the distance from the centre (*"leader"*), at or close to steady state ($10'$ after take-off), for a fleet of 37 drones and 3 different values of the *delta-V* parameter and for a target separation of $50m$ (max. velocity $= 10\frac{m}{s}$). The bars indicate the corresponding distribution in a perfect hexagonal mesh. 1000 independent realisations from randomised initial conditions per parameter value.

The results shown in Fig. 8 provide at least two useful pieces of information:

1. The distributed algorithm being used is capable of reliably generating a close approximation of a regular hexagonal lattice, which means that, by positioning the *"leader"* at its centre and letting the swarm self-organise, homogeneous coverage of a region of interest (the area of which is controlled solely by the target separation distance) can be obtained. See Figures 6(B) and 7(A) for an illustration.
2. *Delta-V* has a strong influence on the accuracy and stability of the lattice, with lower values ($5\frac{m}{s^2}$) resulting in the near disappearance of a discrete second and third *"rings"*, replaced by a more diffuse *"cloud"* in the same annular region ($\approx 80 - 160m$). One can also observe that the effect of the acceleration parameter is nonlinear, with the difference between $10\frac{m}{s^2}$ and $15\frac{m}{s^2}$ being much lower than between $10\frac{m}{s^2}$ and $5\frac{m}{s^2}$.

3.3 Direct Control: Summary and Conclusions

The practical implication for the operator of a swarm is that by modulating the value of three global parameters shared by all drones, target separation distance, speed and acceleration, he/she can trigger a variety of collective movement patterns. A high *delta-V* and low target speed will tend to *"lock"* individual units in a regular mesh (in 2D, a hexagonal one) the density of which is controlled by the separation distance. Conversely, a high target speed and low acceleration rate will favour a much more dynamic and less predictable behaviour in which UAVs follow complicated *"orbits"* around the geometric centre of the swarm. Clearly, the former may be suitable for certain mission types (e.g., homogeneous coverage or systematic survey), the latter for others (e.g., escorting a potential target by scouting all possible attack vectors).

Critically, alternating between these two types of collective behaviour can be achieved simply by broadcasting updates to the corresponding parameter values without any need for *"micro-management"* or path-planning.

All this requires adaptivity from the user interface designed to supervise and control the swarm. For example, mission-specific customizations are needed particularly for the acceleration control. Visualisations of the effects of different parameter values are essential for the user to understand their effect in practice. Furthermore, to predict the effects before making the actual changes, simulation capabilities are needed to show the operator *'what if'* type of scenarios in order for optimal decision-making. Finally, to minimise mental workload, the user interface needs to support the operational situations in a sufficient level of abstraction for the human operator.

4 Indirect Control Methods

By indirect control methods, we refer to any algorithmic framework designed to make the swarm work towards achieving a certain goal without any real-time intervention by a human operator being necessary. In that sense, it is analogous to *"management-by-exception"* [10], whereby a system or organisation is capable of operating smoothly by default and only needs new instructions or temporary takeover occasionally, when circumstances or objectives change.

The chosen scenario to illustrate this concept is long-term surveillance, specific cases of which could be patrolling a remote border or protecting a wilderness area by deterring damaging human activity (e.g., poaching or illegal logging). Because such a mission would typically vastly exceed a drone's battery life, this implies the presence of at least one and probably several *"bases"* at which individual units can land and recharge. Simple navigation methods can be used to guarantee that a drone's flight path never exceeds its autonomy and that it can safely land either at its point of origin (round-trip) or at another base located at the end of a one-way flight.

The challenge is elsewhere and consists in finding ways of leveraging cooperative effects to ensure good coverage (i.e., no *"blind spots"*) and avoid duplication of effort (i.e., several drones patrolling the same area simultaneously).

We found that a suitable way to achieve cooperation with minimal need for explicit coordination and communication is to let all drones access a shared real-time simulation of the world, also known as so-called *Digital Twin* (DT), and modify it to exchange relevant information (a method of interaction known as *stigmergy*). However, the focus of the present paper being user interfaces and methods for orchestrating or choreographing swarms, we will not go into a detailed description and performance analysis here. This is the subject of a separate publication by the same authors team [27].

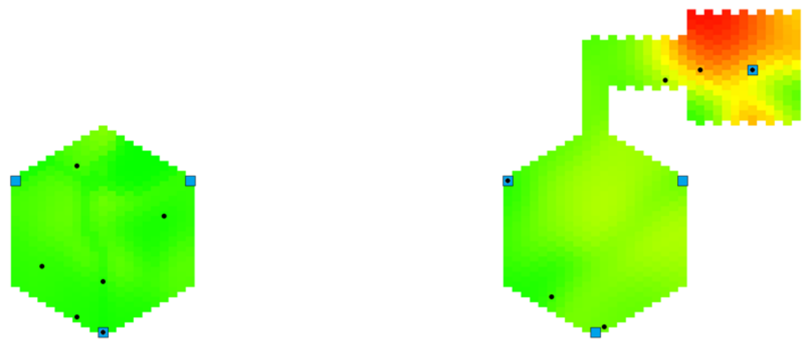

Fig. 9. Images from the user interface for a simulation of the proposed *control through area designation* paradigm. White indicates areas not under consideration, the colour spectrum indicates how close/recently a particular zone has been patrolled by at least one drone (green ≈ nearby and/or a short time ago, red ≈ far and/or a while ago. The panel on the left shows some initial area for the swarm to operate in. Shown are three bases (square, blue) and six drones (one of which is currently at a base). The panel on the right shows the same area after it has been extended towards the North East, using a *"drag-click"* tool of the interface.

Our opinion is that, in the long-term surveillance scenario, interaction between the swarm and its human master should also take the form of modifications to the DT. For instance, the obvious first step would likely consist in designating the default target area for which the user wants the swarm to *"take responsibility"*. The easiest and most intuitive way to do so is to access the real-time simulation through a graphical interface in which standard *"paint"* or *"click-and-drag"* functions can be used to seamlessly add or subtract geographic zones to or from the region of interest. This is illustrated by Figures 9 and 10.

Using this functionality, the human user has added a rectangular area to the North East as well as a connecting corridor. While the number of drones has remained the same, an additional location for a base has also been designated in the middle of the new area.

In essence, when preparing its upcoming flight plan, a departing drone accesses the DT and uses the information it contains to determine the most desirable patrol route at this point in time, based on a suitable utility function.

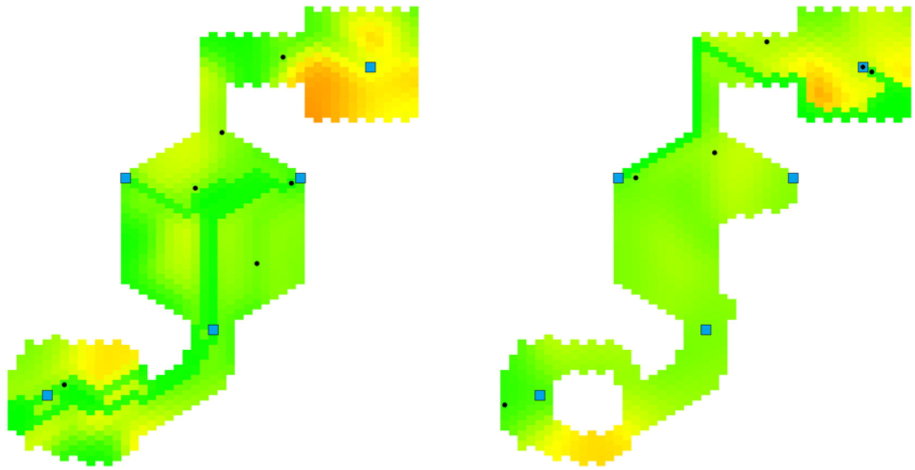

Fig. 10. *Control through area designation*: the area from Figure 9 (right panel) is amended even further. The panel on the left further modifies the area using *"paint" toadd* an irregular region to the South-West. The panel on the right shows the area after using *"ctrl-paint" to remove* certain areas from the swarm's *"territory"*.

In the surveillance scenario, the objective is to ensure that no part of the designated area of interest (the boundaries of which can be modified at runtime as shown in Figures 9 and 10) remains unobserved for too long, which is represented in the DT by a *"heat-map"*. This heat-map indicates the concentration of a diffusing virtual signal that is created when no drone is present and is removed when visited. So in order to maximise its contribution to the swarm's mission, a departing unit only has to *"climb"* the gradient to ensure that it is heading toward an area that needs inspection.

Accordingly, a human operator can also influence the swarm priorities by manipulating the signal. Removing a region from the swarm's territory is effectively done by removing and stopping production of the signal in the corresponding area. Symmetrically, adding a region takes the form of starting production of the signal where it was absent before. Obviously, dropping a large *"quantity"* of the diffusing signal at a given location (*"honeypot"*) will result in a gradient that will attract drones to this area, a method that can be used to *"bias"* the swarm in favour of patrolling a particular zone. This can be regarded as the indirect control equivalent of an explicit command to *"go there"*.

5 Comparison and Conclusion

To conclude this paper we reflect on what as discussed and proposed. In line with the rest of the article, we are considering two main view points: that of someone concerned primarily with the control aspect of a swarm and the accompanying considerations regarding control, from a human factor point of view.

From the control methods point of view, perhaps the most useful lesson to be learned is that it is relatively straightforward to foster the emergence of desirable collective behaviour by leveraging self-organisation. This paradigm is exceedingly familiar in complexity science and its potential will come as no surprise to experts in the field, from Physics to Biology, but it still appears under-exploited in technology and engineering in general and in distributed robotics in particular. This could be about to change with the increasingly realistic prospect of drone swarms performing various useful tasks in the physical world. Quite simply, if we want to make the most of this opportunity to delegate various complex tasks to machines, self-organisation and collective artificial intelligence will cease to be an optional design choice to become a practical requirement.

It should however be remembered that *undesirable* emergent properties are as common as desirable ones, which is why the principled study of global dynamics in a system governed by local rules is extremely important, not least because they can be counter-intuitive or at least not obvious. We encountered this kind of unexpected behaviour during the course of the present investigation, which led to some adjustments. For instance, in the direct control algorithm, the decision to always include the *"leader"* in the trio of influencers was made after it was noticed that not doing so could result in the formation of isolated clusters. After modification, this effect is still present but now transient because the corresponding configuration is usually unstable (see intermediate stages in Figure 5). Retrospectively, this could have been easily anticipated, but some emergent dynamics are more subtle.

This simulation-based *"trial-and-error"* approach, where candidate algorithms are tested, modified, then re-evaluated using quantitative and qualitative measures of performance, may appear ad-hoc in nature, but it is surprisingly effective. The reason is that, after being trained in complex systems modelling, a human designer can gain an intuitive understanding of what particular technique could solve a particular problem, much as a skilled craftsman knows what tool is most suitable to perform a certain task. We expect this type of expertise to become increasingly in demand over the coming decade, not only at design time but also for everyday operations. Indeed, the end user of a drone swarm will rely upon the same familiarity with emergent properties and self-organisation to orchestrate collective behaviour effectively and efficiently, particularly in the face of unexpected events or circumstances. This has clear implications for the design of user interfaces that are fit for purpose.

From a human factors perspective, some conclusions can be drawn related to both control methods. It is clear that direct control requires more active and hands-on user operation than indirect control. The direct control mode of operation may allow the operator to stay better *"in-the-loop"* as regards to what is happening with the swarm at each moment.

However, for efficient control of large swarms in complex operations, indirect control is needed to mitigate, for example, potential human operator workload issues. One clear future research issue in indirect control is in how to achieve the necessary level of operator situation awareness about what the swarm is currently doing. In addition, questions such as how to design the user interface to support operator SA at an appropriate level, how to support the calibration of user trust in the user interface, and how to make the indirect drone swarm control to be an engaging activity are relevant. When combining the direct and indirect control modes to a single user interface, the issues of mode transitions (see e.g., [28]) and handover implementation from indirect to direct control become essential. Clearly, more human factors research is needed in this challenging area.

References

1. Adams, J.A.: Cognitive Task Analysis for Unmanned Aerial System Design, pp. 2425–2441. Springer Netherlands, Dordrecht (2015). https://doi.org/10.1007/978-90-481-9707-1_26, https://doi.org/10.1007/978-90-481-9707-1_26
2. Al-Kaff, A., Madridano, A., Campos, S., García, F., Martín, D., de la Escalera, A.: Emergency support unmanned aerial vehicle for forest fire surveillance. Electronics **9**(2) (2020). https://doi.org/10.3390/electronics9020260, https://www.mdpi.com/2079-9292/9/2/260
3. Bilimoria, K.D., Johnson, W.W., Schutte, P.C.: Conceptual framework for single pilot operations. In: Proceedings of the International Conference on Human-Computer Interaction in Aerospace. HCI-Aero '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2669592.2669647, https://doi.org/10.1145/2669592.2669647
4. Boubeta-Puig, J., Moguel, E., Sánchez-Figueroa, F., Hernández, J., Preciado, J.C.: An Autonomous UAV Architecture for Remote Sensing and Intelligent Decision-making. IEEE Internet Computing **22**(3), 6–15 (Jun 2018). https://doi.org/10.1109/MIC.2018.032501511
5. van den Broek, B., van der Velde, J., van den Baar, M., Nijsten, L., van Heijster, R.: Automatic threat evaluation for border security and surveillance. In: Counterterrorism, Crime Fighting, Forensics, and Surveillance Technologies III. vol. 11166, pp. 113 – 122. Int. Society for Optics and Photonics, SPIE (2019). https://doi.org/10.1117/12.2532308, https://doi.org/10.1117/12.2532308
6. Cabon, P., Coblentz, A., Mollard, R., Foulliot, J.: Human vigilance in railway and long-haul flight operation. Ergonomics **36**(9), 1019–1033 (1993). https://doi.org/10.1080/00140139308967974, https://doi.org/10.1080/00140139308967974, pMID: 8404831
7. Calero, D.P., Peyaud, A., van der Wal, D., van 't Hof, J., Hakkesteegt, H., Vink, R., Bovenkamp, E.G.P., van Antwerpen, G.: Low altitude remote sensing. In: Meynart, R., Neeck, S.P., Shimoda, H., Habib, S. (eds.) Sensors, Systems, and Next-Generation Satellites XII. vol. 7106, pp. 353 – 363. International Society for Optics and Photonics, SPIE (2008). https://doi.org/10.1117/12.799809, https://doi.org/10.1117/12.799809
8. Calhoun, G.L., Draper, M.H.: Display and Control Concepts for Multi-UAV Applications, pp. 2443–2473. Springer Netherlands, Dordrecht (2015). https://doi.org/10.1007/978-90-481-9707-1_88, https://doi.org/10.1007/978-90-481-9707-1_88
9. Coops, N., Goodbody, T., Cao, L.: Four steps to extend drone use in research. Nature **572**, 433–435 (08 2019). https://doi.org/10.1038/d41586-019-02474-y
10. Cummings, M.L., Mitchell, P.J.: Predicting controller capacity in supervisory control of multiple UAVs. IEEE Transactions on Systems, Man,

and Cybernetics - Part A: Systems and Humans **38**(2), 451–460 (2008). https://doi.org/10.1109/TSMCA.2007.914757
11. Cummings, M., Mastracchio, C., Thornburg, K., Mkrtchyan, A.: Boredom and Distraction in Multiple Unmanned Vehicle Supervisory Control. Interacting with Computers **25**(1), 34–47 (01 2013). https://doi.org/10.1093/iwc/iws011, https://doi.org/10.1093/iwc/iws011
12. Endsley, M.R.: Toward a theory of situation awareness in dynamic systems. Human Factors **37**(1), 32–64 (1995). https://doi.org/10.1518/001872095779049543, https://doi.org/10.1518/001872095779049543
13. Endsley, M.R.: Automation and situation awareness. Automation and human performance: Theory and applications **20**, 163–181 (1996)
14. Fairley, R.E., Thayer, R.H.: The concept of operations: The bridge from operational requirements to technical specifications. Ann. Softw. Eng. **3**, 417–432 (Jan 1997)
15. Fuchs, C., Borst, C., de Croon, G.C.H.E., van Paassen, M.M.R., Mulder, M.: An ecological approach to the supervisory control of UAV swarms. International Journal of Micro Air Vehicles **6**(4), 211–229 (2014). https://doi.org/10.1260/1756-8293.6.4.211, https://doi.org/10.1260/1756-8293.6.4.211
16. Goodrich, M.A., Cummings, M.L.: Human Factors Perspective on Next Generation Unmanned Aerial Systems, pp. 2405–2423. Springer Netherlands, Dordrecht (2015). https://doi.org/10.1007/978-90-481-9707-1_23, https://doi.org/10.1007/978-90-481-9707-1_23
17. Hildmann, H., Kovacs, E.: Review: Using unmanned aerial vehicles (UAVs) as mobile sensing platforms (MSPs) for disaster response, civil security and public safety. Drones **3**(3) (2019). https://doi.org/10.3390/drones3030059, https://www.mdpi.com/2504-446X/3/3/59
18. Hildmann, H., Kovacs, E., Saffre, F., Isakovic, A.F.: Nature-inspired drone swarming for real-time aerial data-collection under dynamic operational constraints. Drones **3**(3) (2019). https://doi.org/10.3390/drones3030071, https://www.mdpi.com/2504-446X/3/3/71
19. Hocraffer, A., Nam, C.S.: A meta-analysis of human-system interfaces in unmanned aerial vehicle (UAV) swarm management. Applied Ergonomics **58**, 66 – 80 (2017). https://doi.org/https://doi.org/10.1016/j.apergo.2016.05.011, http://www.sciencedirect.com/science/article/pii/S0003687016300989
20. Karvonen, H., Heikkilä, E., Wahlström, M.: Safety challenges of AI in autonomous systems design – solutions from human factors perspective emphasizing AI awareness. In: Harris, D., Li, W.C. (eds.) Engineering Psychology and Cognitive Ergonomics. Cognition and Design. pp. 147–160. Springer International Publishing, Cham (2020)
21. Kim, L.H., Drew, D.S., Domova, V., Follmer, S.: User-defined swarm robot control. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. p. 1–13. CHI '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3313831.3376814, https://doi.org/10.1145/3313831.3376814
22. Koch, W., Mancuso, R., West, R., Bestavros, A.: Reinforcement learning for UAV attitude control. ACM Trans. Cyber-Phys. Syst. **3**(2) (Feb 2019). https://doi.org/10.1145/3301273, https://doi.org/10.1145/3301273
23. Lee, J.D., See, K.A.: Trust in automation: Designing for appropriate reliance. Human Factors **46**(1), 50–80 (2004). https://doi.org/10.1518/hfes.46.1.50_30392, https://doi.org/10.1518/hfes.46.1.50_30392, pMID: 15151155

24. Lotufo, M.A., Colangelo, L., Perez-Montenegro, C., Canuto, E., Novara, C.: UAV quadrotor attitude control: An ADRC-EMC combined approach. Control Engineering Practice **84**, 13–22 (2019). https://doi.org/https://doi.org/10.1016/j.conengprac.2018.11.002, https://www.sciencedirect.com/science/article/pii/S0967066118305148
25. Parasuraman, R., Sheridan, T.B., Wickens, C.D.: Situation awareness, mental workload, and trust in automation: Viable, empirically supported cognitive engineering constructs. Journal of Cognitive Engineering and Decision Making **2**(2), 140–160 (2008). https://doi.org/10.1518/155534308X284417, https://doi.org/10.1518/155534308X284417
26. Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J.F., Breazeal, C., Crandall, J.W., Christakis, N.A., Couzin, I.D., Jackson, M.O., Jennings, N.R., Kamar, E., Kloumann, I.M., Larochelle, H., Lazer, D., McElreath, R., Mislove, A., Parkes, D.C., Pentland, A.S., Roberts, M.E., Shariff, A., Tenenbaum, J.B., Wellman, M.: Machine behaviour. Nature **568**(7753), 477–486 (Apr 2019). https://doi.org/10.1038/s41586-019-1138-y, https://doi.org/10.1038/s41586-019-1138-y
27. Saffre, F., Hildmann, H., Karvonen, H., Lind, T.: Drones: New Developments and Environmental Applications, chap. Self-swarming for Multi-Robot Systems deployed for Situational Awareness. Remote Sensing/Photogrammetry, Springer (2021)
28. Sarter, N.B., Woods, D.D.: How in the world did we ever get into that mode? mode error and awareness in supervisory control. Human Factors **37**(1), 5–19 (1995). https://doi.org/10.1518/001872095779049516, https://doi.org/10.1518/001872095779049516
29. Star, S.L.: This is not a boundary object: Reflections on the origin of a concept. Science, Technology, & Human Values **35**(5), 601–617 (2010). https://doi.org/10.1177/0162243910377624, https://doi.org/10.1177/0162243910377624